

2.8 CM/CCM

Suppressor models countermeasures (CM) and counter-countermeasures (CCM) using a system type known as Disruptor. Disruptor systems can reduce the effectiveness of sensor and communications systems.

As with weapon systems, disruptor systems can be modeled implicitly or explicitly. The inclusion of IMPLICIT-DETAIL or EXPLICIT-DETAIL in the DISRUPTOR-CHARACTERISTICS table for the disruptor determines which type of modeling is used.

Implicit disruptors can play a part in detection calculations for tracking sensors. For a particular tracking sensor chance, if the signal-to-noise level meets the required threshold and the target contains an implicit disruptor, then a random draw is made from a uniform distribution and compared to the value of the sensor's EFF-BURST-CM-PROB data item. If the draw is less than the input value, then the tracking sensor will lose lock and start coasting. This type of disruptor is one way in Suppressor to model such things as chaff and flares.

Explicit disruptor systems, which use engineering equations to determine their effects, are further categorized in several ways. They can be reactive, meaning that they can be turned on and off and can focus spots of energy at specific sensor and communications transmitters; or they can be non-reactive meaning that they simply emit non-directional energy once they are turned on. Any sensor or communications receiver within the MAX-RNG of the disruptor can potentially be affected by a non-reactive disruptor.

Reactive explicit disruptors can be defined as constant power per spot (CONST-PWR/SPOT) or average power per spot (AVG-PWR/SPOT) in the DISRUPTOR-CHARACTERISTICS table. CONST-PWR/SPOT indicates that the maximum power of the disruptor system will be used on each spot focused at a target transmitter, while AVG-PWR/SPOT will cause the maximum power of the disruptor to be proportionally distributed across all focused spots.

Finally, explicit disruptors can be categorized as NOISE or PULSE. This distinction is described in the following two functional element descriptions. For either to have an effect on sensing and communications events, the sensors and/or transmitters must have the name of the disruptor in their SNR-JMR-INTERACTIONS and/or COMM-JMR-INTERACTIONS tables, respectively. Also, a line of sight must exist between the disruptor and the transmitter and the receiver frequency must be in the interval defined by the DISRUPTOR-FREQ-LIMITS disruptor data item.

2.8.1 Functional Element Design Requirements

The design requirements for the CM/CCM functional element are:

- a. Allow the user to define any number of sensor and communications countermeasure systems on any player.
- b. Provide two levels of detail for countermeasure systems: explicit sensor and communications CM (for signal-to-noise plus jamming calculations) and implicit sensor CM (for probabilistic loss of lock by a sensor tracker).

- c. Provide a level of detail for explicit countermeasure systems representation that is consistent with sensor and communications modeling. For example, the model should use asymmetric antenna patterns, frequency dependent gains, and terrain masking effects.
- d. Provide the capability to allow for an optional dimension in the kill probability table to account for countermeasure effects versus a weapon that cannot be adequately represented with implicit or explicit sensor countermeasures.
- e. Use relative geometry for explicit countermeasure systems interactions with sensors and communications devices. For sensors, the interactions will be on a sensor chance basis while for communications, the interaction will be on a message basis.

2.8.2 Functional Element Design Approach

Design Element 8-1: Noise ECM

This functional element describes noise jamming in Suppressor. A disruptor (jammer) is identified as a noise jammer by the inclusion of the NOISE keyword in the system's DISRUPTOR-CHARACTERISTICS input table.

The presence of noise jamming can affect any sensing or communications chance in Suppressor. For each sensing and communications chance, the total noise from all noise jammers is considered in the signal-to-noise calculation. This total will be the jammer power used in the signal-to-noise calculation if it exceeds that emitted by any PULSE jammers present (see the next functional element description).

Design Element 8-2: Deception ECM

This functional element describes pulse (deception) jamming in Suppressor. A disruptor (jammer) is identified as a pulse jammer by inclusion of the PULSE keyword in the system's DISRUPTOR-CHARACTERISTICS input table.

The presence of pulse jamming can affect any sensing or communications chance in Suppressor. For each sensing and communications chance, the greater of:

- a. the sum of all PULSE jammers with AVG-PWR/SPOT distribution, and
- b. the single most powerful PULSE jammer with CONST-PWR/SPOT distribution

is considered in the signal-to-noise calculation. If the greater of (a) and (b) exceeds the total contribution of the NOISE jammers (see previous functional element) then it will be the jammer power used in the signal-to-noise calculation.

Design Element 8-3: Frequency Adjustment

This functional element describes the ability of a player to change its radar and communications transmitter frequencies due to the presence of jamming and then for the jammer to adjust to the new frequency.

Radar and Communications Receivers

Suppressor gives the radar or communications system “operator” the ability to change transmission frequencies when he recognizes that his system is being jammed. The system is deemed to be jammed if the received jammer noise exceeds the system’s receiver noise (input item RCVR-NOISE) by some input threshold.

Before any frequency changes can take place, several data items must be included in the input. In either the radar or communications case, there must be one or more alternative frequencies (ALT-FREQ:) in the SDB SYSTEM: entry for the affected receiver.

For sensors, the items J/N-NOISE-OPERATOR-THRESHOLD and J/N-PULSE-OPERATOR-THRESHOLD must be present. These are the threshold values for determining whether the sensor receiver is being jammed by noise jammers and/or pulse jammers, respectively (see the discussion in functional elements 5.1 and 5.2 for calculating received jammer power). If either of these thresholds are exceeded, then the radar is considered jammed. Suppressor will choose the next alternate frequency to use and will schedule an event to complete the frequency change at a time equal to the current game time plus the value of the radar transmitter input CHANGE-FREQUENCY-DELAY.

For communications systems, the item J/N-COMM-OPERATOR-THRESHOLD provides the threshold value. In this case, only noise jamming is considered. If the threshold is exceeded, then Suppressor will choose the next alternate frequency to use and complete the frequency change at a time equal to the current game time plus the value of the SDB data item CHANGE-FREQ-DELAY associated with the network of the affected communications receiver.

Jammers

A jammer may have the ability to adjust its spot frequencies against radars and/or communications receivers which have just changed their frequencies. If the data item TIME-REACT-FREQ-CHANGE is present in the jammer’s CAPABILITY, then the jammer will change its spot center frequency (to the new frequency of the jammed system if it is between the UPPER-FREQ-LIMIT and LOWER-FREQ-LIMIT in the DISRUPTOR-FREQ-LIMITS TDB data item; otherwise it will be to the nearer of the UPPER-FREQ-LIMIT or LOWER-FREQ-LIMIT). The length of time to make its adjustment is equal to the TIME-REACT-FREQ-CHANGE item.

2.8.3 Functional Element Software Design

Noise ECM Module Design

The formula used for calculating the power from a jammer is found in routine JAMCAL and is described below:

Power from one jammer is:

$$P_j = (P_x * G * K) / R^2$$

where: $P_x = P_o * B_r * P / (u - 1)$

and: $G = 10^{[(G_j + G_r + L)/10]}$

and: $K = (c/)^2 / (4)^2$

and: $L = L_j + L_r + A_{\text{atm}}$

where:

A_{atm}	Atmospheric attenuation (dB) from TRANSMISSION-LOSS
B_r	Bandwidth of receiver (Hz) from RCVR-BANDWIDTH
c	Velocity of light (m/sec)
	Receiver frequency (Hz) from XMIT-FREQ or SDB FREQ:
f_l	Lower frequency (Hz) of jammer or spot
f_u	Upper frequency (Hz) of jammer or spot
G	Sum of gains and losses
G_j	ANTENNA-PATTERN gain (dB) for jammer transmitter
G_r	ANTENNA-PATTERN gain (dB) for sensor or communications receiver
K	Constant (m^2)
L	Total losses (dB)
L_j	Jammer internal loss (dB) from INTERNAL-LOSS
L_r	Receiver internal loss (dB) from INTERNAL-LOSSES in DETECTION-SENSITIVITIES
P	Receiver polarization factor from POLARIZATION-EFFECTS
P_j	Power received from a jammer (W)
P_x	Power from jammer (W)
P_o	Power output (W) of jammer, equals MAX-POWER-OUT if using CONST-PWR/SPOT MAX-POWER-OUT divided by number of spots if AVG-PWR/SPOT
R	3-dimensional range (m) between jammer and receiver

The design of routine JAMCAL is as follows:

```

*begin logic to perform jammer calculations:
*initialize variables;
*initialize jammer masking entry list;
*loop, until all entry types for all tree levels checked:
  *loop, while specific jammers that can affect receiver:
    *when receiver within jammer frequency limits:
      *determine jammer modulation;
      *when there are any spots:
        *loop, while spots and appropriate one not found:
          *when this spot covers the frequency:
            *set coverage flag and frequency limits;
            *determine if spot intentionally on emitter;
          *end of test for coverage in frequency.
        *end of loop for spots.
      *end of test for any spots.
    *when receiver within spot limits:
      *invoke logic to get present position of jammer;
      *calculate range from jammer to receiver;

```

```

*when jammer within maximum range:
  *search for entry on jammer masking list;
  *when entry is present:
    *look up triangle edge pointers;
  *end of test for jammer entry.
  *when edge pointers are null and either moving:
    *invoke logic to check for line of sight;
  *end of test for edge pointers or moving.
  *when necessary to create new masking entry:
    *add masking entry to list;
  *end of test for masking entry present.
  *store pointers to edge results if necessary;
  *when line of sight exists:
    *invoke logic to calculate gains;
    *compute bandwidth of spot;
    *when pulse jamming:
      *look up subcarrier bandwidth;
      *calculate number of subcarriers required;
      *calculate effective jammer bandwidth;
    *end of test for modulation.
    *when average power used:
      *invoke logic to count number of spots;
    *end of test for average power.
    *calculate effective power within bandwidth;
    *calculate polarization reduction;
    *look up jammer loss;
    *account for transmission losses;
    *solve radio equation for received power;
    *sum if noise jamming;
    *if pulse jamming:
      *if average power per spot distribution:
        *sum the powers;
      *otherwise, constant power per spot:
        *take the maximum;
      *end of test for jamming type.
    *end of test for line of sight.
  *end of test for jammer within maximum range.
*end of test for being within spot limits.
*end of test for receiver within jammer freq limits.
*end of loop for jammers.
*end of loop for address tree levels.
*loop, while jammer masking entries on old list:
  *when masking entry was not used:
    *recycle this unneeded entry;
  *otherwise, keep it for next time around:
    *add masking entry to new list;
  *end of test for unneeded masking entry.
*end of loop for old masking entries.
*store pointer to new, merged list;
*end of logic for JAMCAL.

```

Deception ECM Module Design

The formula for calculating power from a PULSE jammer is the same as that for NOISE, with one exception. The calculation of jammer bandwidth depends on the value of the SUBCARRIER-BANDWIDTH entry in the disruptor input data.

The respective calculations of jammer bandwidth are as follows:

NOISE jammer bandwidth:

$$(u - 1)$$

where:

- f_l Lower frequency (Hz) of jammer or spot
- f_u Upper frequency (Hz) of jammer or spot

PULSE jammer bandwidth:

$$(S * N)$$

where: $N = \text{MAX}(1, \text{INT}((f_u - f_l)/S - 1))$

and:

- S SUBCARRIER-BANDWIDTH from jammer data
- N number of subcarriers

The following is the design of routine JAMCAL, with the pulse-dependent code highlighted:

```
*begin logic to perform jammer calculations:
*initialize variables;
*initialize jammer masking entry list;
*loop, until all entry types for all tree levels checked:
  *loop, while specific jammers that can affect receiver:
    *when receiver within jammer frequency limits:
      *determine jammer modulation;
      *when there are any spots:
        *loop, while spots and appropriate one not found:
          *when this spot covers the frequency:
            *set coverage flag and frequency limits;
            *determine if spot intentionally on emitter;
          *end of test for coverage in frequency.
        *end of loop for spots.
      *end of test for any spots.
    *when receiver within spot limits:
      *invoke logic to get present position of jammer;
      *calculate range from jammer to receiver;
      *when jammer within maximum range:
        *search for entry on jammer masking list;
        *when entry is present:
          *look up triangle edge pointers;
        *end of test for jammer entry.
        *when edge pointers are null and either moving:
          *invoke logic to check for line of sight;
        *end of test for edge pointers or moving.
        *when necessary to create new masking entry:
          *add masking entry to list;
        *end of test for masking entry present.
        *store pointers to edge results if necessary;
        *when line of sight exists:
          *invoke logic to calculate gains;
          *compute bandwidth of spot;
          *when pulse jamming:
            *look up subcarrier bandwidth;
            *calculate number of subcarriers required;
            *calculate effective jammer bandwidth;
          *end of test for modulation.
        *when average power used:
          *invoke logic to count number of spots;
        *end of test for average power.
        *calculate effective power within bandwidth;
```

```

        *calculate polarization reduction;
        *look up jammer loss;
        *account for transmission losses;
        *solve radio equation for received power;
        *sum if noise jamming;
    *if pulse jamming:
        *if average power per spot distribution:
            *sum the powers;
        *otherwise, constant power per spot:
            *take the maximum;
        *end of test for jamming type.
    *end of test for line of sight.
    *end of test for jammer within maximum range.
    *end of test for being within spot limits.
    *end of test for receiver within jammer freq limits.
    *end of loop for jammers.
    *end of loop for address tree levels.
    *loop, while jammer masking entries on old list:
        *when masking entry was not used:
            *recycle this unneeded entry;
        *otherwise, keep it for next time around:
            *add masking entry to new list;
        *end of test for unneeded masking entry.
    *end of loop for old masking entries.
    *store pointer to new, merged list;
    *end of logic for JAMCAL.

```

Frequency Adjustment Module Design

The code which determines whether a radar is jammed and should change its frequency is found in routine OBSRDR:

```

.
.
.
*when operator thinks the radar is being jammed:
    *set operator thinks the radar is being jammed flag;
    *when there are alternate frequencies:
        *when not currently changing frequency:
            *get next available frequency;
            *schedule change frequency event;
            *write out "starts to change freq" message;
        *end of test if not currently changing frequency.
    *end of test for alternate frequencies.
*otherwise, operator thinks the radar is not jammed:
    *clear operator thinks the radar is being jammed flag;
*end of test if operator thinks the radar is jammed.
.
.
.

```

The code which determines whether a communications receiver is jammed and should change its frequency is found in routine YAKKER:

```

.
.
.
*when signal quality was not high enough:
    *schedule sender to notice this;
    *when operator thinks the commo is being jammed &
    * there are alt. frequencies:
        *get next available frequency;
        *write out "starts to change freq" message;

```

```

        *invoke logic to adjust jammer spots focused at rx;
    *end of test if operator thinks the commo is being
    * jammed & alt. frequencies.
    .
    .
    .

```

The code which determines which jammers should adjust their spot center frequencies to match the new frequency of a radar or communications receiver is found in routine SPOTAD:

```

*begin logic to find and adjust jammer spots:
  *initialize variables;
  *when receiver has jammer interactions:
    *loop, until all address node levels checked:
      *loop, while more (jil) jammer interaction types:
        *loop, while jammer entries (jil) that affect rcvr:
          *get jammer frequency limits;
          *when jammer can dynamically follow target emitter:
            *determine player owning the jammer;
            *loop, while more spots from this jammer:
              *loop while more jammer buffers from spot:
                *when spot directed at this emitter:
                  *ensure new spot freq within limits;
                  *schedule spot frequency change;
                  *end of test if spot directed at emitter.
                *end of loop while more buffers for spot.
              *end of loop for spots.
            *end of test if jammer can dynamically follow.
          *end of loop while jmr entries (jil) that affect rcvr.
        *end of loop while more (jil) jammer interaction types.
      *end of loop until all address node levels checked.
    *end of test if receiver has jammer interactions.
  *end of logic to find and adjust jammer spots.

```

2.8.4 Assumptions and Limitations

CM systems are primarily modeled as disruptors which add noise to the receiver signal-to-noise equation. CM systems which utilize deceptive techniques such as range gate or velocity gate stealing must currently be modeled as noise disruptors.

2.8.5 Known Problems or Anomalies

None.